lecoeur electronique

LECOEUR ELECTRONIQUE
19, rue de Courtenay
45220 CHUELLES - FRANCE -
Tel : +33 (0)2 38 94 28 30
Fax : +33 (0)2 38 94 29 67

mailto:info@lecoeur-electronique.com
http://www.lecoeur-electronique.com

Rédacteur	*S. BERTOLOTTO*
Date	*13/04/2004*
Emplacement de création	*Labview\Dll_Ap_int_usb\ Manuel d utilisation dll ap_int_usb_fr rév7.doc*
Documents liés	

révision 7 le 01 mars 2005

Labview\Dll_Ap_int_usb\V16\dll \Ap_int_usb.dll

1 Menu

2 I/O

INPUT	OUTPUT
	Code
ProductNumber	
ChannelNumber	
Function	
In1	
In2	
In3	
In4	
In5	
In6	
	Out1
	Out2
	Out3
	Out4
	Out5
	Out6
Graphe (tableau)	
Graphe lenght	

3 declaration

*int32 ApintUsb(uInt32 ProductNumber, uInt32 ChannelNumber, char Function[], float64 In1, float64 In2, float64 In3, float64 In4, float64 In5, float64 In6, int16 *Out1, int16 *Out2, int16 *Out3, int16 *Out4, int16 *Out5, int16 *Out6, int32 Graphe[], int32 *len)*

Note: C standard call

4 description

First of all: ProductNumber and ChannelNumber are 2 parameters which must be set each time to :

- *0 = USBox*
- *1 = USBoxS*
- *0..7 = For an 8 multi channel system**

Note:

- *Up case letter and space are important*
- ***Code** return 1 is function is recognized else 0. It's not an error code*

Menu

Function	***Description***	***Input***	***Output***
"Init usb", "Usb init", "init usb", "usb init"	Initialise the USB2 connection		
"Channel", "channel"	Set the current channel	*	
"Id code", "id code"	Hardware's code reading		Out # 1
"Prf", "prf"	PRF adjustment	In # 1 = PRF (kHz)	
"Echo-start", "echo-start"	Echo-start inhibition	In # 1 = Echo-start position (µs) In # 2 = Echo-start width (µs) = In # 1 = 0 => echo-start OFF	
"Pulse delay", "pulse delay"	Delay	In # 1 = Pulse delay (µs)	
"Filter/Mode", "filter/mode"	Filter setting and single/double crystal Mode	In # 1 = filter In # 2 = 1 single crystal / 0 double crystal	
"Gain", "gain"	Gain adjustment	In # 1 = gain (dB)	
"Voltage", "voltage"	Pulse voltage adjustment	10 ≤ In # 1 ≤ 230 (V)	
"Width", "width"	Pulse width adjustment	0 ≤ In # 1 ≤ 255	
"Echo-start mode", "echo-start mode"	echo-start on/off + polarity	In # 1 = 0 negative wave / 1 positive wave In # 2 = 0..100 (%) echo-start threshold	
"Scale delay", "scale delay"	Delay adjustment	In # 1 = delay (µs)	
"Wave", "wave"	Select Wave to control	In # 1 = 0 rectified / 1 negative / 2 positive	
"Gate position", "gate position"	Gate position	In # 1 = Gate number 1..3 In # 2 = Gate position (µs)	
"Gate width", "gate width"	Gate width	In # 1 = Gate number 1..3 In # 2 = Gate width (µs)	
"Gate hight", "gate hight"	Gate height	In # 1 = Gate number 1..3 In # 2 = Gate height (%)	
"Relays", "relays"	Alarm on/off	In # 1 = bit0 : Gate1 / bit1 : Gate2 / bit2 : Gate3 bit value : 0 alarm on appear. / 1 disapp	
"Alarm filter", "alarm filter"	Strike before alarm	In # 1 = Gate number 1..3 In # 2 = Number of shots before alarm	
"Measures", "measures"	Gates measures No alarm = 0		Out # 2= alarme1 (MSB), amplitude1 (LSB) Out # 3 = alarme2 (MSB), amplitude2 (LSB) Out # 4 = alarme3 (MSB), amplitude3 (LSB) Out # 5 = distance1 (step of 12.5ns) Out # 6 = distance2 (step of 12.5ns) Out # 1 = distance3 (step of 12.5ns)

"A-scan", "Ascan", "a-scan", "ascan"	Get a-scans coming from the 12bit analog to digital converter	In # 1 = 0 HF / 1 A-scan In # 2 = Retentivity display (0..255) In # 3 = Number of samples In #4 = A-scan wave : 0 full rectified / 1 negative / 2 positive	Graphe Out#1=1 timeout no A-scan
"Help", "help"	Function list		Graphe U8 (ASCII code)
"Version", "version"	Software Version		
"Load configuration", "load configuration"	Load a configuration file coming from the directory c:\saphirp\ustcad	In # 1 = Number of channel	
"Getcounter", "getcounter"	Read counter #x	In#1 : counter# (1-2)	Out#1 : LSB Out#2 : MSB
"Setcounter", "setcounter"	Set counter #x to 8388608	In#1 : counter# (1-2)	
"Resetcounter", "resetcounter"	Reset counter #x	In#1 : counter# (1-2)	
"Counter divider", "counter divider"	Divide the frequency of the counter #x	In#1 : counter# (1-2) In#2 : divider	
"A-scan counter"	Get a frame of several A-scan synchronized on counters		Graphe U8 Out#1=number of samples stored
"SamplingFreq/Mode", "samplingfreq/mode", "samplingfreqmode"	Change the sampling frequency	In#1 : 0=160 1=80 2=40 3=20 ~~4=10MHz~~ In#2 : 0=strapped 1=disconnected (pulse/receiver) If frequency is different of 80MHz, filter is set to broadband	
"Scale A-scan counter"	Define the number of samples of the frame	In#1 : Step of ***25ns*** Note : It can be different of the A-scan *§ A-scan counter function*	
“Dac”, “”dac”	Program the DAC curve	In#1 : 0=Dac OFF / 1=Dac ON Graphe *§Dac curve function*	

5 A-scan counter function

*This function read a frame of **several** A-scan between 2 dll calls. These A-scan are sampled on **8bit** and stored in real time inside a fifo of **512kO**.The frequency of the storage depend of the counters frequency which can be divided by the “counter divider” function.*
*USBOXS stores the current A-scan when coder frequency divide by “counter divider”, generates a pulse which is **synchronized on the next PRF.***
That’s why it’s important to define “Prf”, upper then coder frequency

- *The frame begins by 2 header bytes FF (255). All others are always different of FF*
- *Coders values are stored on 3 bytes with MSB of each byte, set to **0** (It's a base of 128)*

  *Coder#1 value = (counter line3) + (counter line4 * 128) + (counter line5 * 16384)*
- *The frame length is defined by the "Scale A-scan counter" function.*

  *Note: "Scale A-scan counter" is set by step of **25ns** but A-scan length is returned by step of the period defined by "SamplingFreq/Mode" (6.25ns, 12.5ns, 25ns, 50ns)*

  Example: You want a sampling of 10µs (=10000ns) at 80Mhz (12.5ns)
  - *"Scale A-scan counter" : In#1 = 10000ns/25ns = **400***
  - *One frame length of "A-scan counter" : 10000ns/12.5ns = **800***

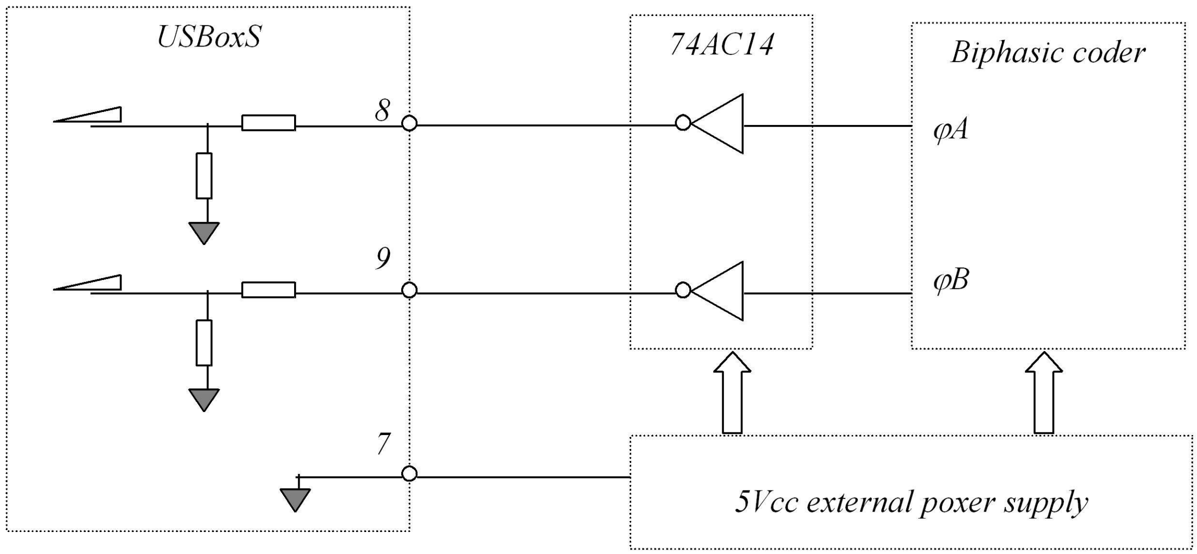
- ***The 19 1st bytes are replaced by headers, coders, amplitudes etc**.* Menu

The 18 first bytes of the A-scan are :

1- FF (header)
2- FF (header)
3- counter #1 LSByte
4- counter #1
5- counter #1 MSByte
6- counter #2 LSByte
7- counter #2
8- counter #2 MSByte
9- amplitude gate #1
10- amplitude gate #2
11- amplitude gate #3
12- distance gate #1 LSB
13- distance gate #1 MSB
14- distance gate #2 LSB
15- distance gate #2 MSB
16- distance gate #3 LSB
17- distance gate #3 MSB
18- number of samples LSB
19- number of samples MSB
20- sample1 ←1st sample of As-scan
21- sample2
22- .
23- .
24- .
25-

6 Coders interface

USBoxS manage 2 biphasic coders (HEDS-5701#A00 for example) but to adapt TTL signals to 2.5Vcc an internal resistors divider is wired (220/330Ω).
LECOEUR LECTRONIQUE company suggest to add a Schmitt trigger (74AC14) between USBoxS and coders with an external power supply.



Menu

7 Dac curve function

The USBoxS DAC curve is 166µs depth.
It programmable by 256 step of 650ns.

To program it, send 256 gain values (dB) through "Graphe" and set 1 in "In#1"

Take care, if you want to adjust the gain after programming a Dac curve, don't use "Gain" function but reprogram a new Dac curve with an offset.
Using "Gain" function while a Dac curve will set it off !

Over the 166µs, the gain of "Graphe[0]" is reprogrammed.

The 1st point of the curve, "Graphe[0]", correspond to the transmitter shot (Original sync)

Menu